Due in class Mon 4/30

This will be a closed-book/closed-notes exam.

1. The standard SVM problem formulation:

$$\begin{aligned} \min_{\mathbf{w}} \ & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{n=1}^{N}\xi_n \\ \text{s.t.} \ & y^n(\mathbf{w}^T\mathbf{x}_n + b) + \xi_n \geq 1, \\ & \xi_n \geq 0, \ \forall n. \end{aligned}$$

penalizes datapoints that are on the wrong side of the margin boundary with a linear term ξ_n in the objective function. Suppose that we wish to strictly require each datapoint to fall on the right side of the margin boundary by eliminating the slack variables ξ_i:

$$\begin{aligned} \min_{\mathbf{w}} \ & \frac{1}{2}\mathbf{w}^T\mathbf{w} \\ \text{s.t.} \ & y^n(\mathbf{w}^T\mathbf{x}_n + b) \geq 1 \end{aligned}$$

a) Find the dual of this optimization problem.

$$L(w, b, \alpha) = \frac{1}{2}w^Tw + \sum_n \alpha_n [1 - y_n(w^Tx^{(n)} + b)]$$

$$\frac{\partial L}{\partial w} = 0 = w - \sum_n \alpha_n y_n x^{(n)} \qquad w = \sum_n \alpha_n y_n x^{(n)}$$

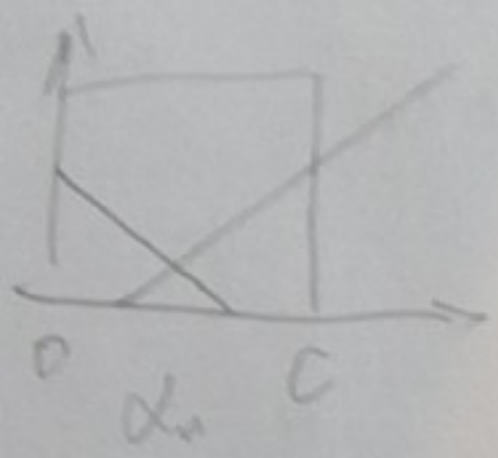
$$\frac{\partial L}{\partial b} = 0 = -\sum \alpha_n y_n$$

$$g(\alpha) = \frac{1}{2}\sum_n \sum_m y_n y_m \alpha_n \alpha_m x^{(n)T} x^{(m)} + \sum_n \alpha_n - \sum_n \alpha_n y_n \sum_m \alpha_m y_m x^{(n)T} x^{(m)}$$

$$= -\frac{1}{2}\sum_n \sum_m y_n y_m \alpha_n \alpha_m x^{(n)T} x^{(m)} + \sum_n \alpha_n$$

$$\alpha_n \geq 0$$

b) Explain in words how the dual differs from the dual of the original formulation with slack variables. How will this change the optimization procedure in the dual?



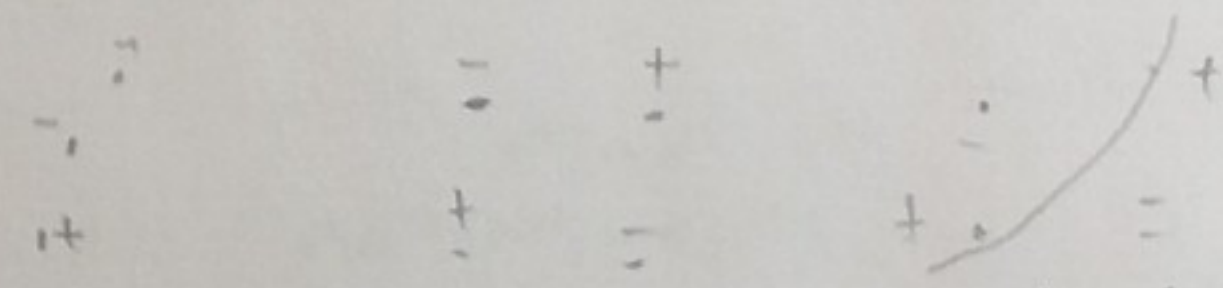
c) The formulation without slack variables has no solution if the training data are not linearly separable. However, a kernel function can be used to address this problem. How can we choose a kernel such that the optimization problem is guaranteed to have a solution? What are possible drawbacks of your method in comparison the the original SVM formulation with slack variables?

$$K(x', x'') = e^{-c\|x' - x''\|^2}$$

2. (a) What is the VC Dimension of circles in the plane? For this concept class, a concept is a point x a radius r; all points within distance r of x are positive instances.

VCD ≥ 3 some 3 points shattered

VCD < 4 no 4 points



(b) We proved in class that if $VCD(C) < \infty$, then $\Pi_C(m)$, the number of ways of classifying m points, grows at most polynomially in m. Show that the converse is also true: if $\Pi_C(m)$ grows at most polynomially, then $VCD(C) < \infty$.

$\Pi_C(m) = O(m^d) \Rightarrow VCD(C) < \infty$

$\exists m' \ \Pi_C(m') < 2^{m'} \Rightarrow VCD(C) < \infty$

$VCD(C) = \infty \Rightarrow \Pi_C(m) = 2^m$

3. Principal Components Analysis minimizes the error of the reconstructed points $\hat{x}$ over a data set of N points $x^{(1)}, \ldots, x^{(N)}$:

$$E_M = \frac{1}{2} \sum_{n=1}^{N} \|x^{(n)} - \hat{x}^{(n)}\|^2$$

Show that this is equivalent to maximizing the probability of the data points when each point's distribution is specified by a probabilistic error term ϵ added to the reconstructed point:

$$x^{(n)} = \hat{x}^{(n)} + \epsilon^{(n)}$$

$$P(x^{(n)} \mid \hat{x}^{(n)}) = P_\epsilon(\epsilon^{(n)})$$

What probability distribution P_ϵ over ϵ makes this formulation equivalent to the original formulation? You should show that minimizing E_M is equivalent to maximizing $P(x^{(1)}, \ldots, x^{(n)} \mid \hat{x}^{(1)}, \ldots, \hat{x}^{(n)})$, but you do not need to do the minimization/maximization itself.

$\min E_M$

$\min \frac{1}{2} \sum_{n=1}^{N} \|x^{(n)} - \hat{x}^{(n)}\|^2$

$\max P(x^{(1)} \ldots x^{(n)} \mid \hat{x}^{(1)}, \ldots \hat{x}^{(n)})$

$\max \prod_n P(x^{(n)} \mid \hat{x}^{(n)})$

$\max \prod_n P_\epsilon(x^{(n)} - \hat{x}^{(n)})$

$\max \log \prod P_\epsilon(x^{(n)} - \hat{x}^{(n)})$

$\max \sum \log P_\epsilon(x^{(n)} - \hat{x}^{(n)})$

$\min \sum -\log P_\epsilon(x^{(n)} - \hat{x}^{(n)})$

$-\log P_\epsilon(x^{(n)} - \hat{x}^{(n)}) = c\|x^{(n)} - \hat{x}^{(n)}\|^2$

$P_\epsilon(x^{(n)} - \hat{x}^{(n)}) = e^{-c\|x^{(n)} - \hat{x}^{(n)}\|^2}$

$= \frac{1}{Z} e^{-\frac{1}{2}\|x^{(n)} - \hat{x}^{(n)}\|^2}$

$= \mathcal{N}(x^{(n)}; \hat{x}^{(n)}, I)$

4. You are given a directed acyclic graph $G = (V, E)$ with a source node $s \in V$ and a sink node $t \in V$. The graph is a lattice: s is the only node with no incoming edges, and t is the only node with no outgoing edges. You wish to use Gibbs sampling to sample from all paths from s to t with uniform probability. Your advisor proposes the following Gibbs sampling algorithm: assign a variable z_v to each vertex v. The values of z_v range over vertices that are reachable in one step from v: $\{u : (v, u) \in E\}$. Any assignment to all variables z_v specifies a path from s to t: the path is determined by starting at s and following the sequence s, z_s, z_{z_s}, ... The algorithm iterates over vertices v and resamples each variable z_v uniformly at random.

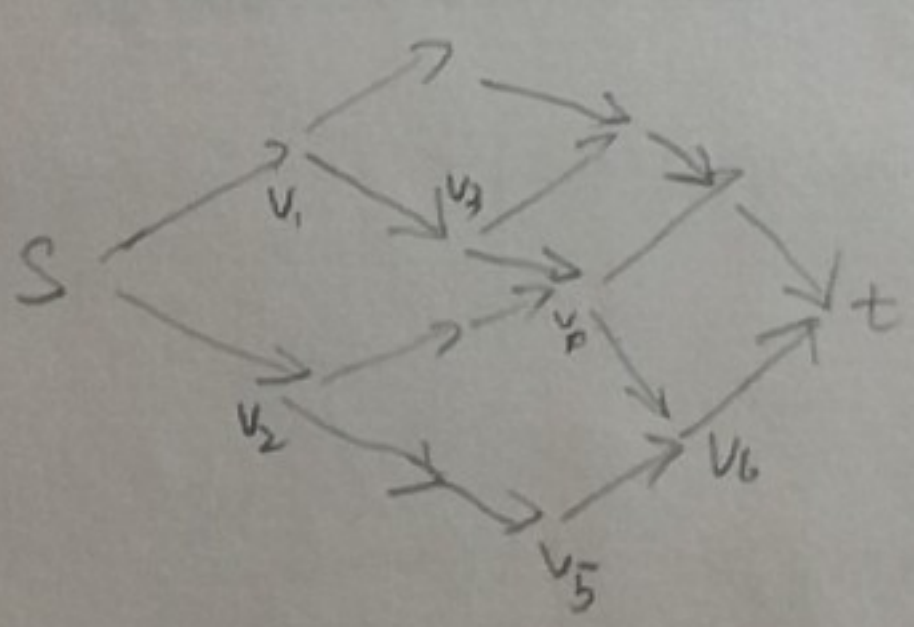
We wish to show that the algorithm converges to a uniform distribution over all paths from s to t.

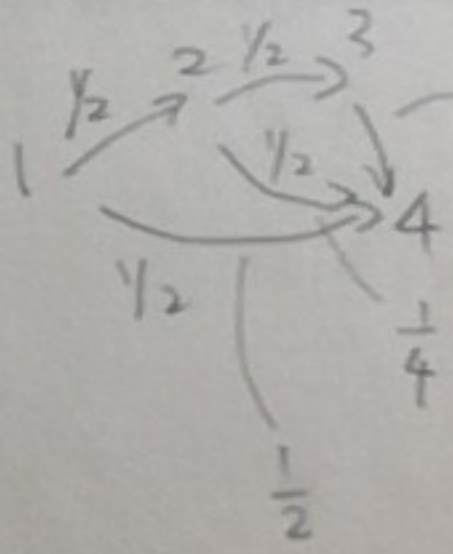
EITHER

Show that the algorithm above converges to the correct distribution by showing that this distribution satisfies detailed balance, and that the chain is aperiodic and irreducible,

OR

Provide an example of a graph for which the algorithm above does not converge to the correct distribution.





$Z_{V_1} = V_3$ $\quad P(Z_1 \ldots Z_n)$

$Z_{V_3} = V_4$ $\quad P(\text{path}) = P(Z_{V_1} = V_3, Z_{V_3} = V_4, Z_{V_4} = V_t)$

$Z_{V_4} = t$ $\quad P(Z_{V_1}, Z_{V_2} \ldots)$

5. The Metropolis-Hastings algorithm discussed in class for MCMC sampling uses a proposal distribution $Q(x';x)$ and accepts the newly proposed state x' with probability:

$$A(x';x) = \min\left(1, \frac{P(x')Q(x;x')}{P(x)Q(x';x)}\right)$$

We showed that this method has the property of detailed balance, which means that the Markov chain looks the same running forwards and backwards.

Suppose that our proposal function is symmetric:

$$\forall x, x'\ Q(x';x) = Q(x;x')$$

and that now we accept the new state x' with probability

$$A(x';x) = \frac{P(x')}{P(x) + P(x')}$$

This is equivalent to "forgetting" which of x and x' is the current state, and choosing between them according to their relative probabilities with respect to P. Does this version of algorithm also satisfy detailed balance? Show why or give a counterexample.

$x' \neq x$

$T(x'|x) = A(x';x)\,Q(x';x)$

$A(x';x) = \frac{P(x')}{P(x)+P(x')}$

$\frac{P(x')}{P(x')+P(x)} Q(x';x) P(x) = \frac{P(x)}{P(x')+P(x)} Q(x;x') P(x')$

$= \frac{P(x')}{P(x')+P(x)} Q(x';x) P(x)$